William Stallings
Computer Organization and Architecture

Chapter 11
Pipelining

Estrutura da CPU

- CPU deve:
  - Buscar instruções
  - Interpretar instruções
  - Buscar dados
  - Processar dados
  - Escrever Dados

Registradores

- CPU necessita de armazenamento temporário
- Denominados registradores
- Quantidade e funções variam por famílias de processadores
- Uma das principais decisões no design
- Primeiro nível da hierarquia de memória

Registradores

- Uso(Propósito) Geral
- Dados
- Endereços
- Códigos Condicionais

Registradores Uso(Propósito) Geral (1)

- Pode ser, realmente, de propósito geral
- Pode ser restrito
- Pode ser usado para dados ou endereços
- Dados
  - Acumulador
- Addressing

Registradores Uso(Propósito) Geral (2)

- Fazê-los de propósito geral
  - Aumenta flexibilidade
  - Aumenta tamanho da instrução e sua complexidade
- Fazê-los especializados
  - Instructions menores (Mais rápidas)
  - Menor flexibilidade

Quantos Registradores de Propósito Geral?

- Entre 8 - 32
- Menos = Mais referência de ,emória
- Mais, não reduz, obrigatoriamente, as referências de memória
- Rever RISC

Qual Tamanho ?

- Grande suficiente para caber um endereço
- Grande o suficiente para caber uma instrução completa
- Frequentemente, é possível combiner 2 registradores para armazenar valores maiores (Não se combina para armazenar instruções)
  - Linguagem C e seus derivados
  - double int a;
  - long int a;

Registradores de Códigos Condicionais

- Conjuntos de bits
  - Ex.: Resultado da última operação
- Pode ser lido implicitamente por programas
  - Ex.: Pular se zero
- Não pode ser atribuído por programas

Resgitradores de Controle e Status

- Program Counter
- Instruction Decoding Register
- Memory Address Register
- Memory Buffer Register

- Revision: O que são esses registradores ?

Modo Supervisor

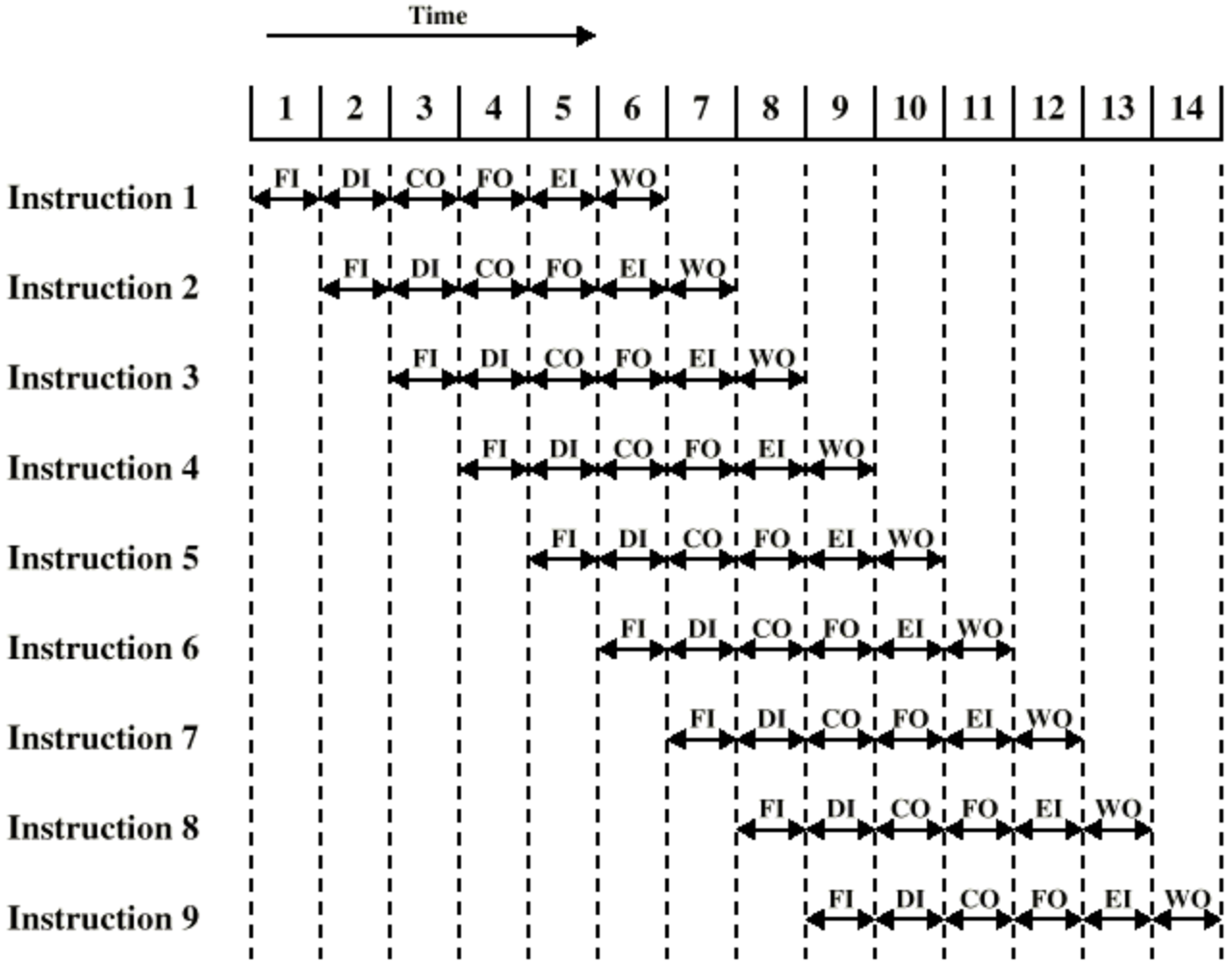
- Intel anel 0
- Modo Kernel
- Permite a execução de instruções que requerem privilégios
- Usados por Sistemas Operacionais
- Não disponiveis para aplicação de usuários

Pipelining

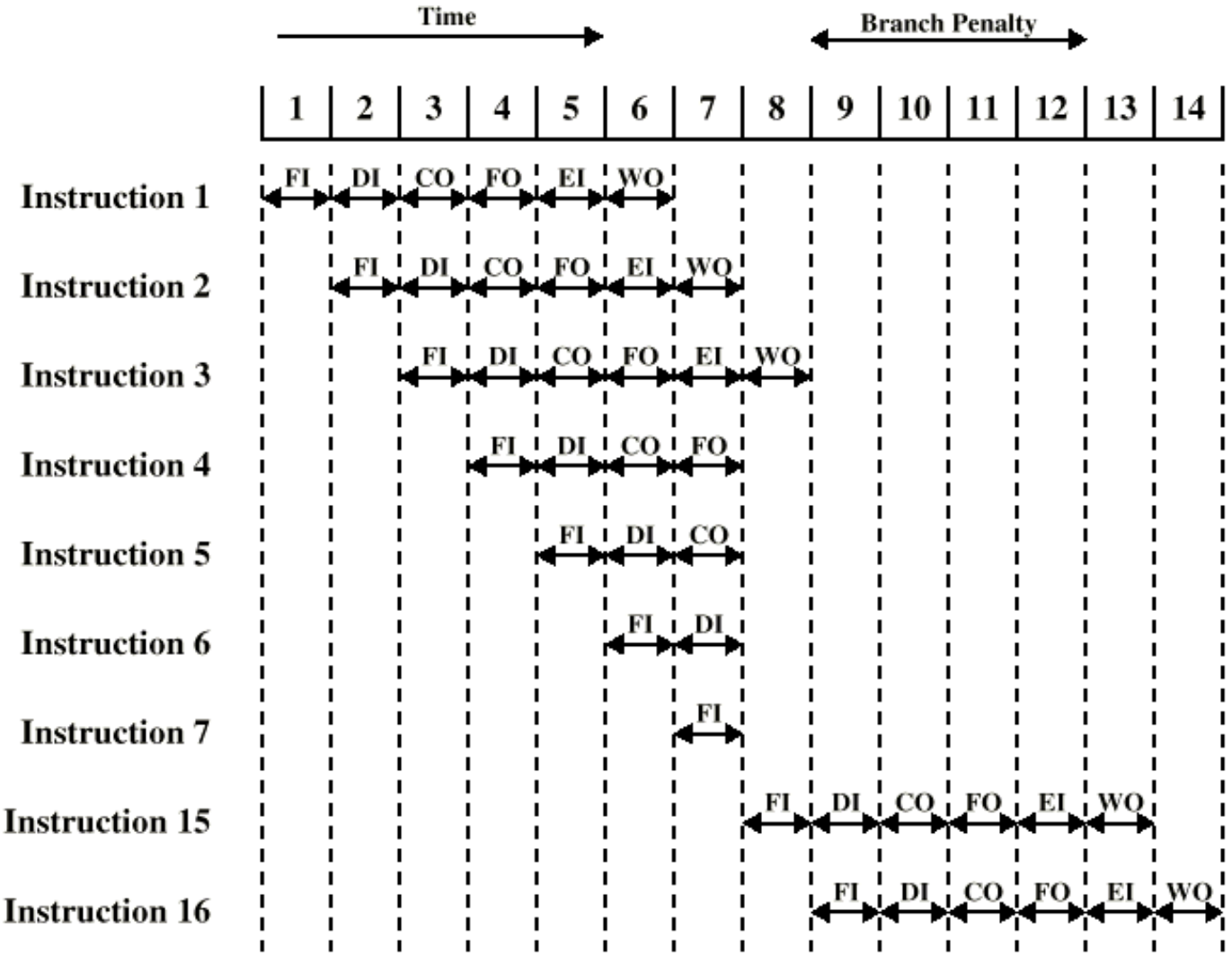
- Buscar Instrução - Fetch instruction (FI)
- Decodificar Instrução - Decode instruction (DI)
- Verificar Endereços - Calculate operands (CO)
- Buscar Dados - Fetch operands (FO)
- Executar Instruções - Execute instructions (EI)
- Escrever Resultados - Write result (WR)

- Sobrepor essas operações

Temporização do Pipeline



Branch em Pipeline



Lidando com Branches

- Multiple Streams (Múltiplos Fluxos)
- Prefetch Branch Target (Pré busca de alvo)
- Loop buffer (Repetição)
- Branch prediction (Previsão de branches)
- Delayed branching (Branches com atraso)

Múltiplos Fluxos

- Ter 2 pipelines
- Transforma cada branch em um pipeline separado
- Usar o pipeline apropriado

- Múltiplos branches levam a mais piplines

Prefetch Branch Target

- O alvo da busca é definido a partir da definição da instrução
- Mantenha o alvo enquanto o branch é executado
- Usado pelo IBM 360/91

Loop Buffer

- Verifica o buffer antes de busca na memória
- Boa estratégia para pequenos laços ou saltos
- Usado pelo CRAY-1

Execução Especulativa

- Execução especulativa consiste em um conjunto de técnicas para antecipar a execução de instruções antes que todas as dependências tenham sido resolvidas.
- A não resolução de todas as dependências pode significar inclusive que instruções são executadas desnecessaria ou erroneamente. Todavia, o ganho representado pela não paralisação do pipeline é comumente maior que o custo de possíveis execuções equivocadas.

Execução Especulativa

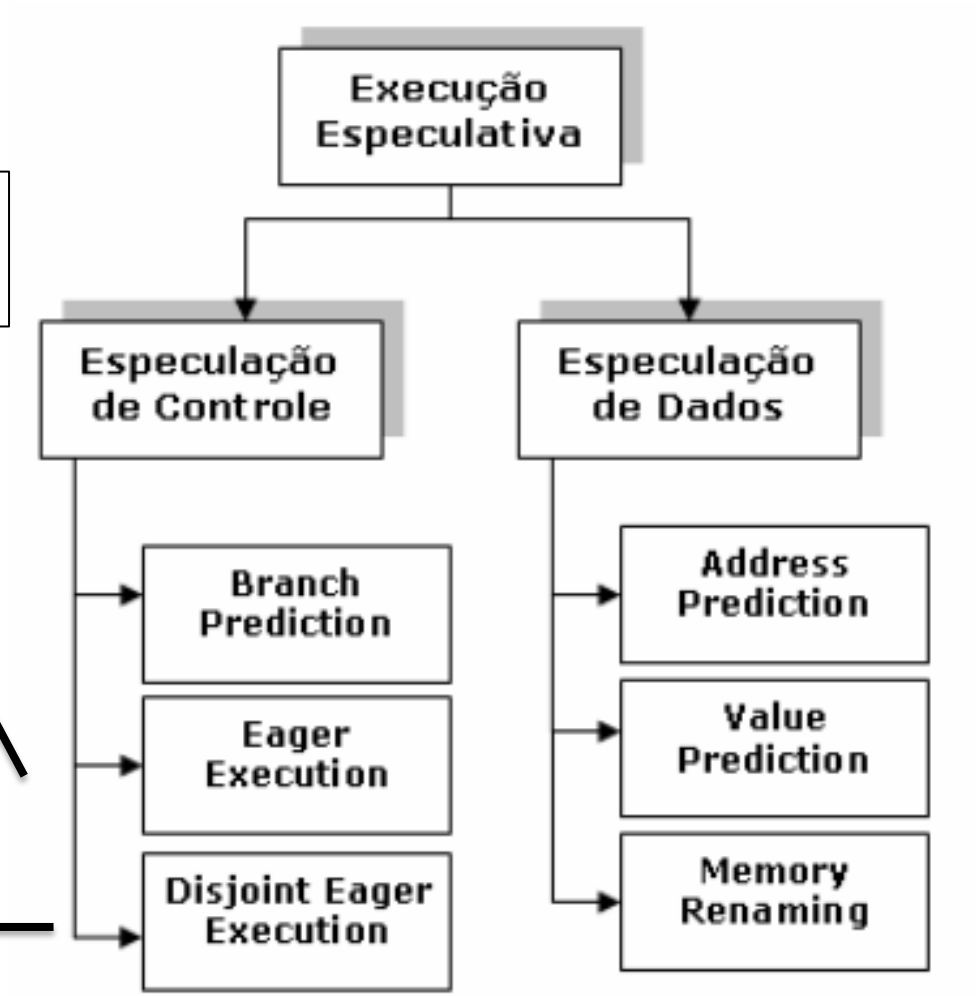
- A antecipação proporcionada pela execução especulativa, na maioria dos casos, é baseada em análise estatística dos resultados previamente obtidos. Essa análise é realizada explorando-se a localidade de valor, espaço e tempo, que é observada na imensa maioria dos softwares desenvolvidos.

Execução Especulativa

- As Dependências de Dados ocorrem quando uma instrução depende de um dado que ainda não foi obtido ou calculado por outra instrução precedente. Principalmente em virtude da latência da instrução load, as dependências de dados consistem num importante gargalo para os processadores com pipeline.
- As seguintes técnicas de execução especulativa procuram reduzir esse problema:
  - Address Prediction: Técnicas que tentam predizer em qual posição de memória dados ou instruções estão armazenados.
  - Value Prediction:Técnicas que procuram predizer qual o valor está armazenado em um registrador ou em um endereço de memória.
  - Memory Renaming:Técnicas que procuram comunicar valores já armazenados para instruções loads.

Execução Especulativa

Nessa técnica todos os possíveis caminhos de execução são testados



Variação da técnica
Eager Execution em que as restrições de disponibilidade de recursos são levadas em consideração para determinação de quais caminhos, selecionados de acordo com a probabilidade de serem tomados, serão executados.

Meltdown & Spectre

- Spectre: As brechas chamadas de Spectre quebram a barreira de isolamento entre aplicações. Potencialmente permitem que uma aplicação maliciosa ”force” outra aplicação a acessar de forma arbitrária segmentos de informações em sua memória, que podem ser visualizadas por um canal paralelo, vazando informações. A vulnerabilidade é baseada em uma falha de design, que explora a técnica de execução especulativa.
- Meltdown: A brecha chamada de Meltdown quebra a barreira de isolamento entre aplicação e sistema operacional. Permite que qualquer aplicação executada pelo usuário acesse a memória central do computador. A palavra meltdown significa derreter, em inglês, em alusão ao fato de que a aplicação pode ignorar a barreira de segurança do processador, que impediria o acesso à memória do kernel dos sistemas operacionais (seja Windows, Linux ou macOS). O kernel possui acesso a praticamente todos os processos de um dispositivo.

Previsão de Branches (1)

- Previsão que nunca ocorrerá
  - Assume que o salto nunca ocorrerá
  - Sempre busca a próxima instrução
  - 68020 & VAX 11/780
- Previsão que sempre ocorrerá
  - Assume que sempre ocorrerá o salto
  - Sempre faz uma busca de alvo

Previsão de Branches (2)

- Previsão por OpCode
  - Algumas instruções, estatísticamente, tem maior probabilidade de que o salto ocorra
  - 75% success
- Taken/Not taken
  - Baseado no histórico
  - Bom para laços

Previsão de Branches (3)

- Branch com atraso
  - Não faz o salto até que seja necessário
  - Rearranja as instruções